REMARKS

Applicants appreciate the time taken by the Examiner to review Applicants' present application. This application has been carefully reviewed in light of the Official Action mailed October 19, 2004. Applicants respectfully request reconsideration and favorable action in this case.

Rejections under 35 U.S.C. § 101

Claim 31 stands rejected under 35 U.S.C. § 101. Claim 31 has been amended in a manner similar to that suggested by the Examiner to include one or more "computer readable media" storing "computer program code". Claims 32-39 have also been amended to reflect the change in the preamble of Claim 31. Applicants believe that these changes also address the Examiner's objections to Claims 32, 33, and 34.

Rejections under 35 U.S.C. § 102

Claims 1-39 stand rejected as anticipated by U.S. Publication No. 20020065962A1 ("Bakke").

Claim 1

Claim 1 recites associating a command with a command identifier. By using a command identifier, the host device can precisely identify any commands that failed in transmission between the host device and the sequential device. This can allow the correct command, or portion of the command, to be retransmitted over a different port.

In rejecting Claim 1, the Examiner cites paragraphs 0036 and 0037 of Bakke, which read:

> [0036] With reference to FIG. 3, there is shown a simplified logic chart of the functions and the device mechanisms preferably embodied in the adapter 140 which are used in accordance with principles of the invention. In block 310, the host processor complex 102 issues a command from the operating system 122, 124 along the host system bus 115 to an adapter 140, preferably an I/O adapter. Typical I/O commands that are issued from the host operating system 122, 124 include READ/WRITE, FORMAT, REASSIGN, READ CAPACITY, etc. The adapter 140 preferably includes a redundancy manager 350 which manifests many of the features of the invention as described herein. The redundancy manager 350 herein could also be implemented by control circuitry through the use of logic gate, programmable logic devices, or

> other hardware components in the adapter in lieu of microcode. Within the adapter 140, microcode or firmware may perform advance function processing as provided, e.g., write caching 340, RAID or other device scheduling 342, and device command processing 344 to, inter alia, build a command in the device language.
>
> [0037] Features of the invention will now be described with respect to the processes of the adapter. The command issued by the host operating system and/or applications 122, 124 may be stored in the adapter's write cache 340, if available. A new command is selected from the write cache 340 according to a command issuance scheme. If there is specialized function processing such as compression, read caching, RAID scheduling, mirroring, etc., those processes occur under the auspices of the device function microcode 342. The device command processing section 344 of the adapter 140 then logically translates the new command into the device language and builds a command the device can interpret. From the device command processing 344, the command issues to the redundancy manager 350, which in accordance with principles of the invention, dynamically determines which physical path will be used for transmission of each command. Once the redundancy manager 350 chooses the physical path, the command is forwarded to a layer of code called the chip encapsulation 346 which sets registers and hardware of the device interface chips 224, 234 for the adapter to actually talk across the physical path to the device itself. From the device interface chips 224, 234, the command is received in the device and the device executes the command. The device then sends a response indicating if the command was successfully executed or if an error or other conditions attach to the response. The response returns to the chip encapsulation code 346 which in turn notifies the redundancy manager 350 and forwards the command response to the device command processing 344. If any error recovery occurs because the command was unable to execute, error recovery may take place in the device command processing 344. The device command processing 344 forwards the response to the host operating system 122, 124.

It is unclear where these paragraphs describe assigning an identification to a command as they simply suggest that the chip encapsulation code receives a response to the command. How a response is correlated to a command is not described (i.e., based on timing or based on some other mechanism). Applicants, therefore, respectfully request that the Examiner more specifically point out where these features of Claim 1 can be found or allow Claim 1.

Claim 2

Claim 2 recites that the system further comprises "a router coupled between the host device and the sequential device, wherein the router is configured to count commands transmitted from the host device to the sequential device and to associate the count with the corresponding command." According to Claim 2, a router counts commands issued by a host device and associates each count with the corresponding command. This can allow the router to determine which command failed or if a command received from the host device is a new

command or is a command sent as part of error recovery. *See* paragraph 0042 of the present application.

In rejecting Claim 2, the Examiner cites paragraph 0039 of Bakke and states that the router is the "redundancy manager". Paragraph 0039 of Bakke reads:

> [0039] In one aspect of the invention, the redundancy manager 350 discovers and resolves all the devices on all physical paths to which it is connected. Although there may be N physical paths to a particular device, the redundancy manager 350 ultimately presents one logical path to the operating system 122 and the device and command functions above the redundancy manager 350 by correlating information from the N paths and resolving existing aliases. The redundancy manager 350 interrogates each physical path and determines the number of active/inactive devices on each path by reading the world wide identification code and/or the vital product data. Using the identification code and/or the vital product data, the redundancy manager 350 then resolve aliases, correlate the separate physical paths to/from each device into one logical path and presents the device to the operating system. Further, the redundancy manager 350 conforms commands on each physical path to the ordering semantics and other requirements of the operating system and maps the command to the physical capabilities of the protocol of the physical path used, for example, the redundancy manager 350 would implement the queue tags of the SCSI architectural model (SAM) protocol.

As an initial matter, it is unclear as to where paragraph 0039 of Bakke teaches that any portion of Bakke counts commands or that a router is configured to count commands transmitted from the host device to the sequential device. Applicants, therefore respectfully request that the Examiner more particularly point out where this feature of Claim 2 can be found in the portions of Bakke cited by the Examiner so that Applicants can more fully respond to the Examiner's rejection.

Additionally, Applicants submit that the redundancy manger of Bakke is part of the host device and not a router coupled to the host device. FIGURE 3 of Bakke shows that the redundancy manger 350 of Bakke is part of an adapter 140 which is connected to host processor complex 104 via a host system bus (see FIGURE 1). The adapter allows the host system to connect to a variety of peripheral devices according to a number of protocols. Thus, the redundancy manager of Bakke is part of the host system adapter that allows a host system to connect to peripheral devices and is not a router coupled to the host device.

Moreover, even if the redundancy manager is a router as asserted by the Examiner, paragraph 0039 discusses how the redundancy manager identifies peripheral devices connected by N paths. There is no teaching or suggestion that, even as a router, the redundancy manager should count commands.

Claim 3

Claim 3 recites that "the host device and the router are configured to begin counting commands . . . ." Counting commands at a router and a host device allows the router to identify a command received over a secondary link as a new command or a previously issued command on a failed link. *See* paragraph 0042.

In rejecting this Claim, the Examiner cites to paragraphs 0037 and 0039, reproduced above. Applicants can not find a reference in these paragraphs to counting commands at a host device and a router. As stated in the discussion of Claim 2, the redundancy manager is part of the host device adapter and not a router. Even if the redundancy manger is considered a router solely for the sake of argument, these paragraphs do not disclose counting commands at that router and counting the commands at the host device. Without a teaching of both counting commands at the host device and counting commands at the router, Bakke does not teach all the features of Claim 3. Applicants therefore request that the Examiner point out where these features of the present invention can be found in Bakke or allow Claim 3.

Claim 10

Claim 10 recites that the router is configured to "identify the command identifier transmitted via the second one of the ports as identical to the command identifier transmitted via the first one of the ports". According to Claim 10, the router receives a command and command identifier transmitted over the second port. The command identifier is identical to the command identifier that the host device initially sent over the first port. Thus, when the host device retransmits a command over the second port, it can use an identical identifier so that the router and/or sequential device can identify the command being received over the second port as the command that previously failed.

In rejecting Claim 10, the Examiner cites paragraph 0039 of Bakke, which is reproduced above. This paragraph describes that the redundancy manager can correlate device identifications between various paths to identify the same device on different paths. This does not appear to address assigning an identification to a command. While the redundancy manager "conforms commands on each physical path to the ordering semantics and other requirements of the operating system and maps the command to the physical capabilities of the protocol of the physical path used, for example, the redundancy manager 350 would implement the queue tags of the SCSI architectural model (SAM) protocol," Applicants are unable to find

any teaching regarding a specific identification for a command in paragraph 0039. Furthermore, Applicants are unable to find a teaching that, when the command is retransmitted over a second port, the command is assigned the same command identification. Therefore, Applicants respectfully request allowance of Claim 10.

Claim 12 and Claim 20

Claims 12 and 20 recite that the host is configured to receive an indication of a number of data bytes received by a router in response to requesting the status of the command. This feature of the claims allows the host to determine how much of the data for a particular command was received. In the realm of sequential target devices, this allows the host device to determine what data has already been written to the sequential media (e.g., tape).

Again in rejecting this claim, the Examiner cites paragraph 0039. Paragraph 0039, appears to disclose interrogating target devices via the various paths to determine information necessary to identify the target devices. Paragraph 0039 does not discuss requesting the status of issued commands or receiving the number of bytes of data received for a particular command. The Examiner further cites paragraph 0054, which reads:

> [0054] In a preferred embodiment, the redundancy manager also has the capability to detect a failed physical path and dynamically reroute a command to a device on a path other than the failed path, called failover. The detection, failover, and recovery from a failed physical path is automatic and happens without host operating system or driver software intervention. With the redundancy manager, moreover, a failed physical path does not result in lost access to a resource, it only reduces the total available bandwidth until the failed physical path is repaired. The redundancy manager will not use the failed physical path until it is repaired. Any operations that were in process using the failed physical path are handled such that the device command processing of the adapter uses the identical error recovery procedures as it did in the non-redundant case. The redundancy manager ensures that the peripheral devices are in the state expected by the device command processing, e.g., an ACA state used by the SAM protocol by, for instance, issuing commands to the peripheral device using a functional physical path to get the peripheral device into the expected state. Recovery of a failed physical path, moreover, occurs automatically and is transparent to the host driver software.

Paragraph 0054 merely suggests that the redundancy manager can detect a failed path and issue commands on a working path to place a peripheral device in a state expected by device command processing. Applicants are unable to find a teaching or suggestion in paragraph 0039 or 0054 that the host can receive an indication of the number of data bytes

received by a router as recited in Claim 12. Applicants therefore request that the Examiner point out where Bakke teaches that the host is configured to receive an indication of a number of data bytes received by a router in response to requesting the status of the command or allow Claims 12 and 20.

Claim 13 and Claim 21

Claims 13 and 21 recite that the host device is configured to request that the error recovery begin with the next data byte following a last received data byte. This allows error recovery to begin error recovery where it left off when the first link failed without having to either write the same data twice to the sequential media or rewind the sequential media to overwrite the portion of data already received.

The examiner cites paragraph 0054 of Bakke in rejecting Claim 13 and Claim 21. Paragraph 0054 simply suggests that some form of error recovery can occur and that the redundancy manager can issue commands to place the peripheral device in a certain state as part of error recovery. However, Applicants are unable to find any teaching or suggestion in paragraph 0054 that the recovery process should include requesting that error recovery begin with a next data byte following a last received data byte. In fact, it appears from paragraph 0055 of Bakke that the entire command is resent for execution without reference to how much of the command was received over the failed link. Accordingly, Applicants respectfully request allowance of Claims 13 and 21.

Claim 23 and Claim 31

Claim 23 recites "associating a first command identifier with a first command; transmitting the first command via a first link; detecting a failure of the first link and transmitting at least a portion of the first command and the first command identifier via a second link." Thus, according to Claim 23, if a first link fails, at least a portion of the first command is along with the first command identifier are transmitted on a second link. Claim 31 includes similar recitations for a computer program product.

Bakke, at paragraphs 0036 and 0037 (reproduced above), on the other hand, teaches that if "any error recovery occurs because the command was unable to execute, error recovery may take place in the device command processing. The device command processing forwards the command to the host operating system." Applicants are unable to find a teaching or suggestion in paragraphs 0036 or 0037 that a command (or portion thereof) should be retransmitted on a second link using the same command identifier that was transmitted with the

command on the first link if the first link fails. Therefore, Applicants respectfully request allowance of Claims 23 and 31.

Claim 24 and Claim 32

Claim 24 recites "wherein associating a first command identifier with a first command comprises the host and the router counting commands transmitted from the host device to the sequential device and associating the count with the corresponding command." Claim 32 includes similar recitations for a computer program product. Applicants are unable to find any reference to a host and a router counting commands as part of associating a command with a command identifier in paragraphs 0036, 0037 or 0039 of Bakke. It is unclear from these paragraphs if or where commands are counted as these paragraphs do not appear to discuss counting commands. Applicants therefore respectfully request that the Examiner more particularly point out where counting commands at a host and a router can be found or allow Claim 24 and Claim 32.

Claim 27 and Claim 35

Claim 27 recites "requesting recovery starting at a next byte following a last byte previously received . . . ." Claim 35 includes similar recitations for a computer program product. In rejecting Claim 27, the Examiner cites paragraphs 0039 and 0054 of Bakke. Again, however, Applicants are unable to find a reference in either of these paragraphs to requesting that error recovery occur beginning at a certain byte. Applicants therefore request that the Examiner more particularly point out where this feature of the present invention can be found or allow Claims 27 and 35.

Claim 28 and Claim 36

Claim 28 recites that "the status indicates a number of bytes of the first command actually received by the router." Claim 36 recites that the status "indicates a number bytes of the first command actually received." In either case, the status indicates how many bytes for a particular command were received. Applicants are unable to find any teaching or suggestion in paragraph 0039 or 0054 cited by the Examiner of a status that includes the number of bytes actually received. Paragraph 0039 appears to discuss determining the peripheral devices connected by various paths. Paragraph 0054 merely discusses that error recovery can occur, but does not appear to teach or suggest that a host should receive a status that "indicates a number of bytes of the first command actually received . . . ." Therefore, Applicants request

that the Examiner point out where the cited references teach these features of the present invention or allow Claims 28 and 36.

Conclusion

Applicants have now made an earnest attempt to place this case in condition for allowance. Other than as explicitly set forth above, this reply does not include an acquiescence to statements, assertions, assumptions, conclusions, or any combination thereof in the Office Action. For the foregoing reasons and for other reasons clearly apparent, Applicants respectfully request full allowance of Claims 1-38. The Examiner is invited to telephone the undersigned at the number listed below for prompt action in the event any issues remain.

The Director of the U.S. Patent and Trademark Office is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully submitted,

Sprinkle IP Law Group
Attorneys for Applicants

John L. Adair
Reg. No. 48,828

Date: January 19, 2005

1301 W. 25th Street
Suite 408
Austin, Texas 78705
Tel. (512) 637-9225
Fax. (512) 371-9088